

7 Claims 18 Drawings

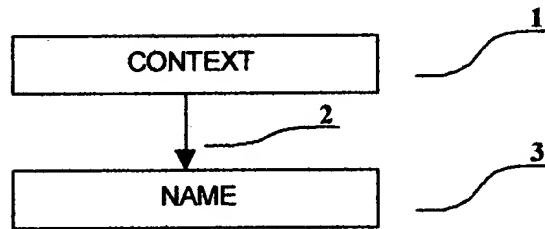


Figure 1. Category Structure

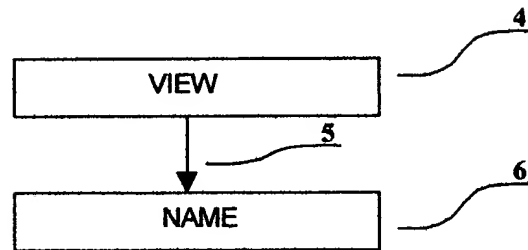


Figure 2. Standard Category Structure for the View Context

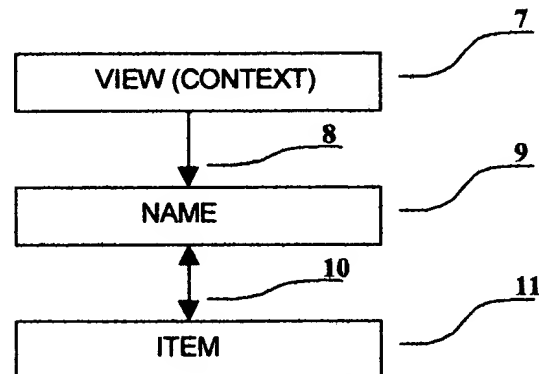


Figure 3. Structure of the Database Information Model.

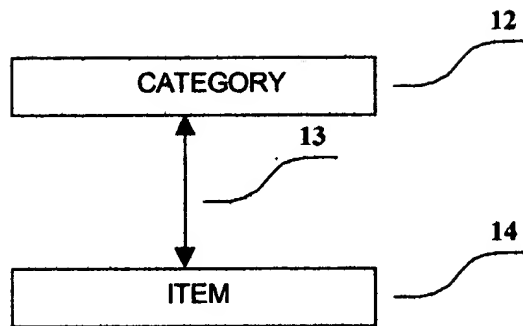


Figure 4. The Simplified Structure of the Database Information Model.

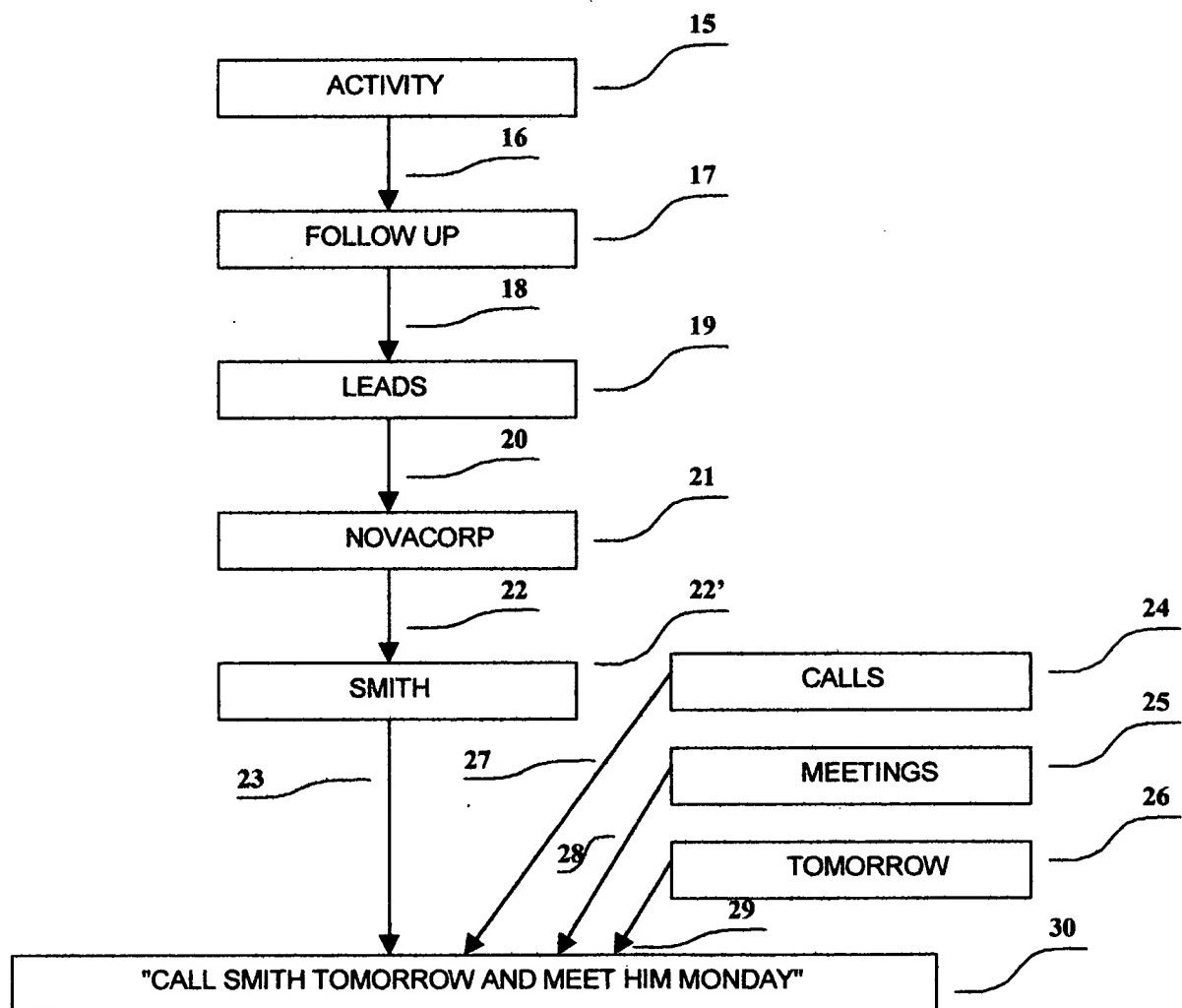


Figure 5. Structure of a sample category classification.

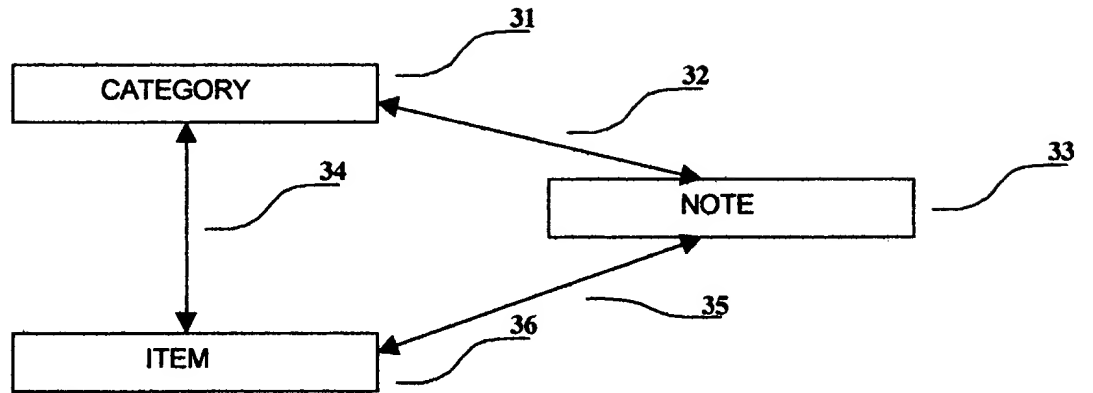


Figure 6. All Elements of the Simplified Database Information Model.

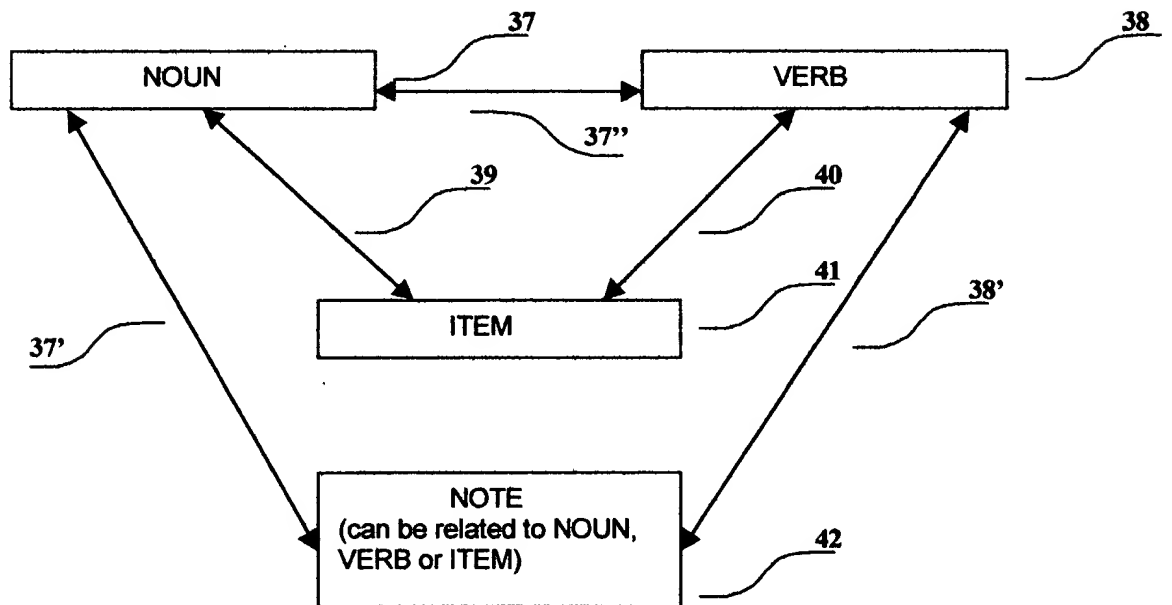


Figure 7. All Elements of the Database.

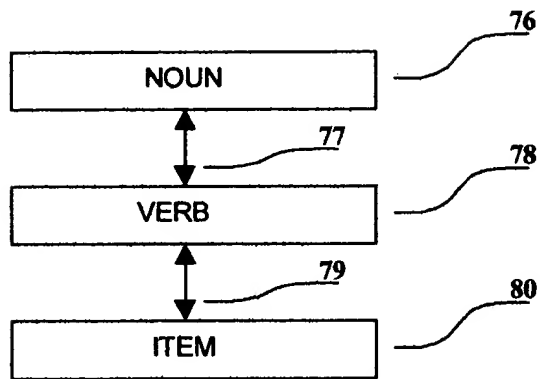


Figure 9. Quantified elementary information.

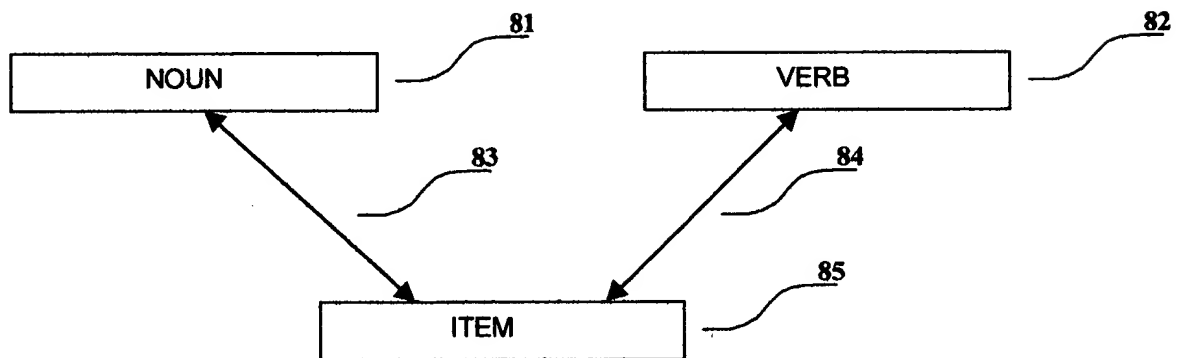


Figure 10. All Elements of the quantified elementary information.

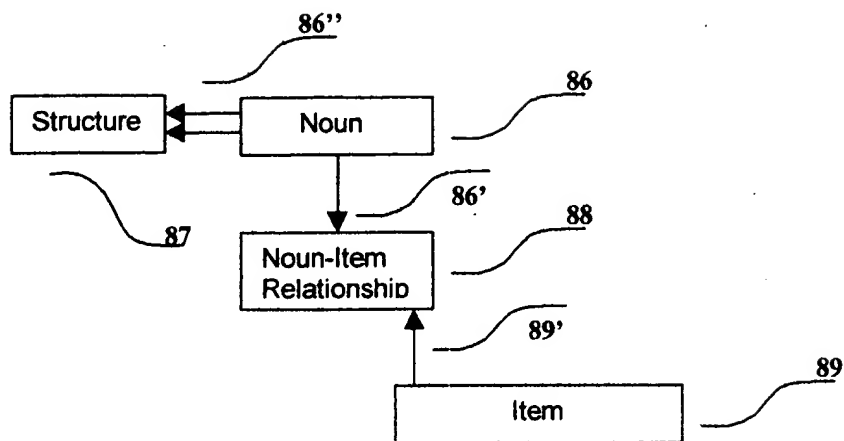


Figure 11. Illustration for the basic retrieval algorithm.

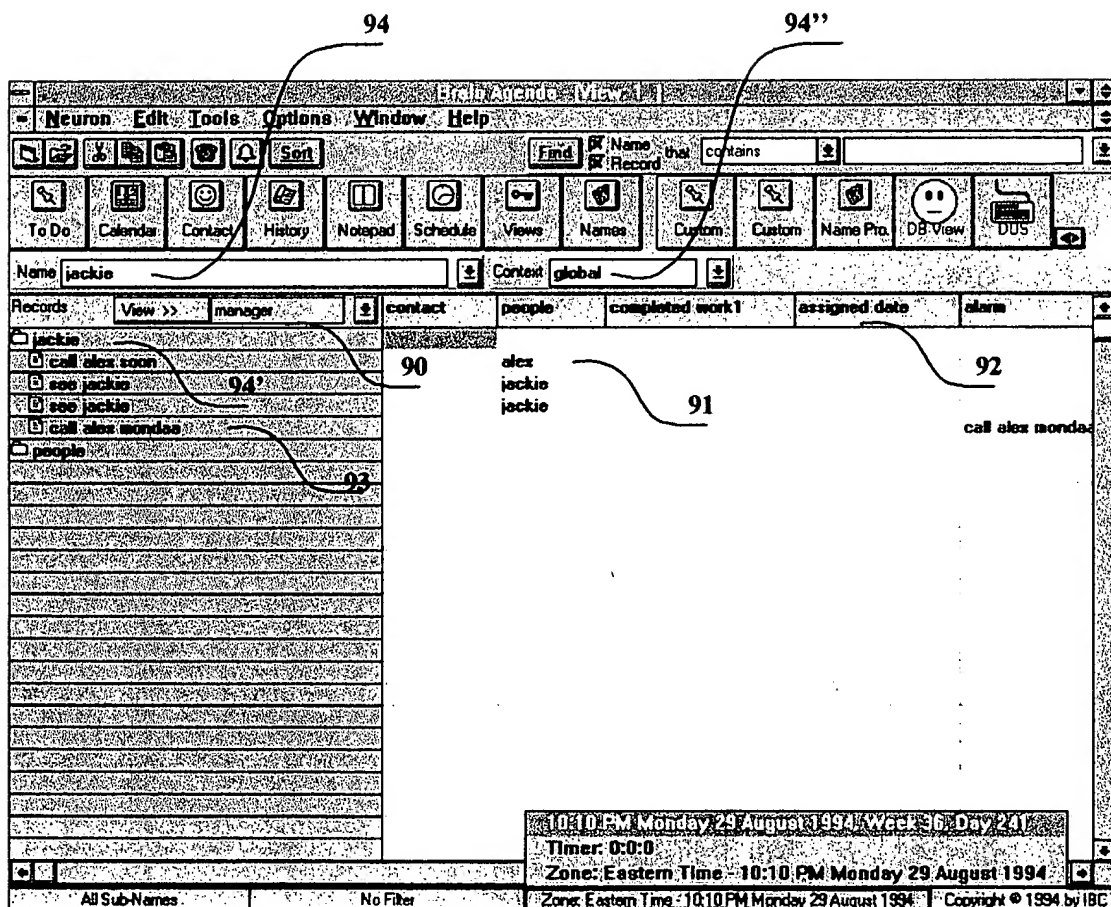


Figure 12. Two dimensional query results of the basic retrieval algorithm.

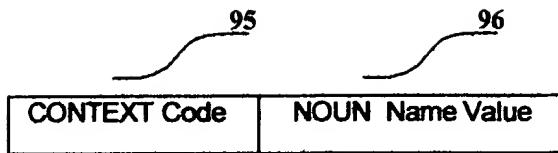


Figure 13. Basic structure of Noun (Verb) record.

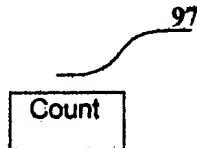


Figure 14. Basic structure of Relationship (or Structure) record containing usage count or certainty factor.

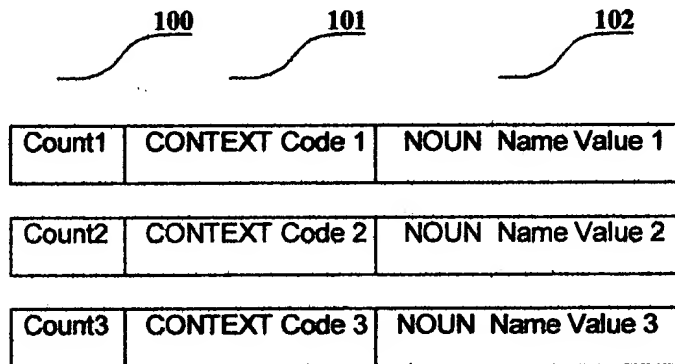


Figure 15. Product of Nouns (Verbs) and Structure of Nouns (Verbs); with three elements.

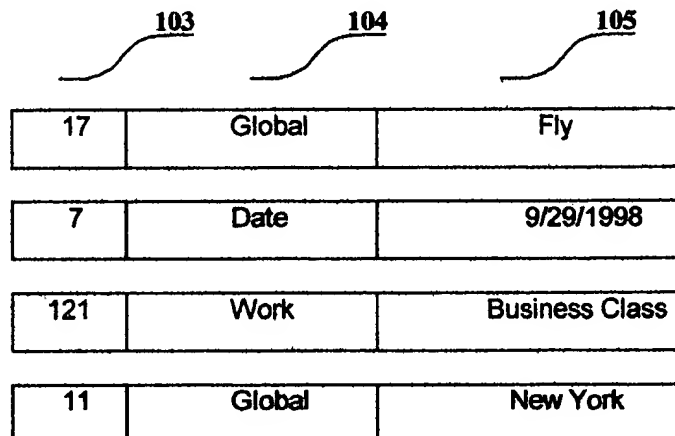


Figure 16. Example of product of Nouns (Verbs) and Structure of Nouns (Verbs); with four elements.

```

/* **** */
/*
/*      BRAIN      Schema for the database BRAIN.
/*      Global schema for every neuron.
/*
/* Identification:
/*      1000-0-00-00-00
/*      ---- - - - -
/*                      neuron  0001/.../1000
/*                      portion  0/1/2
/*                      relation 00/10/.../54
/*                      release  1
/*                      version  1
/*
/*      Portion      1
/*                  0      Abstraction
/*                  1      Reality
/*                  2      Abstraction-Reality relation
/*
/*      Part      11      Noun+Data+Doc
/*                10      Noun
/*                14      Noun-Data
/*                15      Noun-Doc
/*                40      Data
/*                45      Data-Doc
/*                50      Doc
/*
/*      Release      01
/*                  01      Alpha release
/*                  02      Beta release
/*
/*      Version      01
/*                  01      Alpha version
/*                  02      Beta version
/*
/* **** */
/* Module name : Brain Agenda - Personal Information Manager
/*      NEURON_1000
/* **** */
/* Implemented : RAIMA, db_VISTA III
/*
/* Compile type: ddp
/*      def. ddlp -rxbds brain.ddl
/*      -r - report
/*      x - cross reference
/*      b - no alignment
/*      d - dupl. field names
/*      s - case preserve
/* **** */
/* 1. | BRAIN      | 1991.09.01 | New
/* **** */
/* 1000-0-00-00-00      6144*/

```



```

database BRAIN [6144]
{
    data file "F100010.00" contains
/* 1000-0-10-00-00 */

                                noun;

    data file "F100011.00" contains
/* 1000-0-11-00-00 */

                                datar,
                                datar_tabl;

    data file "F100012.00" contains
/* 1000-0-11-00-00 */

                                noun_datar,
                                noun_str,
                                noun_synonim,
                                datar_str,
                                action_before,
                                action_after;

    data file "F100019.00" contains
/* 1000-0-10-00-00 */

                                brain,

/* 1000-0-50-00-00 */

                                note;

    key file "F100010.00K" contains

                                noun.id;

    key file "F100011.00K" contains

                                datar.id;

    key file "F100019.00K" contains

                                note.id;
/*****
/* Sub-schema : BRAIN - NOUN */
/* Description : Noun (Parameter) part of BRAIN */
/*****
/* Record type : brain */
/* Description : Start of the NEURON 1000 */
/*****
record brain
{
    char          db_path [81];          /* Path to database */
    char          db_name [81];          /* name of the db "brain" */
    struct
    {
        long      type_v;                /* noun type, view id */
        char      kname_v [41];          /* noun 40B + 1B null termin*/
        long      subtype_v;            /* noun subtype, def = 0 */
    } id_v;
    char          name_v [256];          /*
struct
{
    long          type_n;                /* noun type, name id */
    char          kname_n [41];          /* noun 40B + 1B null termin*/
    long          subtype_n;            /* noun subtype, def = 0 */
    long          type2_n;              /* noun 2 type, def = 0 */
    char          kname2_n [41];        /* noun 40B + 1B null termin*/
    long          subtype2_n;          /* noun subtype, def = 0 */
} id n;

```

```

        char          name_n [256]; /* */
    long      read_action; /* action on load */
    long      next_1; /* next available ??? */
    long      next_2; /* number for extention */
    long      next_3; /* noun ext.,noun definition*/
    long      value_1 ; /* */
    long      value_2 ; /* */
    long      value_3 ; /* */
    double    double_1; /* */
    double    double_2; /* */
    double    double_3; /* */
    char      reserve_1[41]; /* */
    char      reserve_2[41]; /* */
    char      free[5001]; /* */
}

/*****
/* Record type : noun */
/* Description : names (views,names,contexts) */
/*****
record noun
{
    unique key struct
    {
        long      type; /* noun type, def = 0 */
        char      kname [41]; /* noun 40B + 1B null termin*/
        long      subtype; /* noun subtype, def = 0 */
        long      type2; /* noun 2 type, def = 0 */
        char      kname2 [41]; /* noun 40B + 1B null termin*/
        long      subtype2; /* noun subtype, def = 0 */
    } id;
    char      name[256]; /* 255+1 */
    struct
    {
        long      type_p; /* noun type, pair id */
        char      kname_p [41]; /* noun 40B + 1B null termin*/
        long      subtype_p; /* noun subtype, def = 0 */
    } id_p;
    long      cf; /* certainty factor */
    long      delete; /* */
    long      joint_id; /* neuron||joint long */
    long      read_action; /* action on read */
    double    date_create; /* */
    double    date_when; /* */
    double    date_done; /* */
    double    date_start; /* */
    double    date_end; /* */
    char      short_name [21]; /* 1B null termin*/
    char      cat_type [11]; /* 1B null termin*/
    char      exclusive [2]; /* 1B null termin*/
    char      settings [41]; /* 1B null termin*/
    long      layout_link; /* type of layout for linked note*/
    struct
    {
        long      type_link; /* link to extention which */
        char      kname_link [41]; /* is in note */
        long      subtype_link; /*reserve the range of notes*/
    } id_link;
}

```

```

struct
{
    long    type_note;          /* note id          */
    char    kname_note [41];    /* note name        */
    long    subtype_note;       /* note page        */
} id_note;
    long    position_note;      /*          in document/page */
    char    free_1  [101];
    char    free_2  [101];
    char    reserve_1[21];       /*3 sets person company */
    char    reserve_2[11];       /*          notes (commence) */
    char    reserve_3[11];       /*          notes (commence) */
}

/*****
/* Record type : datar          */
/* Description : records from Brain Agenda          */
*****/
record datar
{
    unique key struct
    {
        long    type;          /* data type, def = 0          */
        char    kname [41];    /* data 40B + 1B null termin*/
        long    subtype;       /* data subtype, def = 0      */
    } id;
    char    name[256]; /* 255+1          */
    long    cf;          /* certainty factor          */
    long    delete;      /*          */
    long    joint_id;     /* neuron||joint          long */
    long    read_action;  /* action on read          */
    double  date_create;  /*          */
    double  date_when;    /*          */
    double  date_done;    /*          */
    double  date_start;   /*          */
    double  date_end;     /*          */
    char    settings [41]; /*          1B null termin*/
}

struct
{
    long    type_note;          /* note id          */
    char    kname_note [41];    /* note name        */
    long    subtype_note;       /* note page        */
} id_note;
    long    position_note;      /*          in document/page */
    long    long_1;            /*          */
    char    reserve_1[11];      /*          */
    char    reserve_2[11];      /*          */
    char    reserve_3[11];      /*          */
    char    reserve_4[11];      /*          */
}

/*****
/* Record type : datar_tabl          */
/* Description : data tables          */
*****/
record datar_tabl
{
    long    elem [120];        /* 120 elements          */
    long    cf;                /* certainty factor          */
}

```

```

        long    delete;                /* */
        double  date_create;            /* */
        long    read_action;            /* action on read */
        double  double_1;               /* */
        char    reserve_1[11];          /* */
        char    reserve_2[21];          /* */
    }
/*****
/* Record type : note */
/* Description : notes (pages ) document */
/*****
    record note
    {
        unique key struct
        {
            long    from;                /* doc id +datar,-name,0-user */
            long    type;                /* from record or name */
            char    kname [41];          /* chapter||paragraph||verse blank*/
            long    subtype;             /* for user=0 */
            long    page_nr;             /* page nr */
        } id;
        char    name [256];             /*
            long    cf;                  /* certainty factor */
            char    chapter [101];       /* left on page */
            char    chapter_1[101];      /* left on page */
            char    chapter_2[101];      /* left on page */
            char    chapter_3[101];      /* left on page */
            char    chapter_4[101];      /* left on page */
            char    chapter_5[101];      /* left on page */
            char    chapter_6[101];      /* left on page */
            long    verse;               /* left on page */
            char    page [5001];         /* page 5001 */
            long    delete;              /*
            long    read_action;          /* action on read */
            char    reserve_1 [11];
            char    reserve_2 [11];
            char    reserve_3 [11];
            char    reserve_4 [11];
        }
/*****
/* Record type : noun_str */
/* Description : structure of the noun */
/*****
    record noun_str
    {
        long    cf;                    /* certainty factor */
        double  date_create;           /*
        long    read_action;           /* action on read */
        double  double_1;              /*
        char    reserve_2[11];         /*
        char    reserve_3[11];         /*
    }
/*****
/* Record type : noun_datar */
/* Description : relation noun - datar */
/*****
    record noun_datar

```

```

{
    long    cf;                /* certainty factor          */
    double  date_create;       /*                            */
    long    read_action;       /* action on read            */
    double  double_1;          /*                            */
    char    reserve_2[11];     /*                            */
    char    reserve_3[11];     /*                            */
}

/*****
/* Record type : action before
/* Description : must belong to the datar before being assigned to
/* the current datar
*****/

record action_before
{
    long    cf;                /* certainty factor          */
    double  date_create;       /*                            */
    long    read_action;       /* action on read            */
    double  double_1;          /*                            */
    char    reserve_2[11];     /*                            */
    char    reserve_3[11];     /*                            */
}

/*****
/* Record type : noun action after
/* Description : is assigned to noun after being assigned to
/* the current noun
*****/

record action_after
{
    long    cf;                /* certainty factor          */
    double  date_create;       /*                            */
    long    read_action;       /* action on read            */
    double  double_1;          /*                            */
    char    reserve_2[11];     /*                            */
    char    reserve_3[11];     /*                            */
}

/*****
/* Record type : noun_synonim
/* Description : all synonyms for a noun
*****/

record noun_synonim
{
    long    cf;                /* certainty factor          */
    double  date_create;       /*                            */
    long    read_action;       /* action on read            */
    double  double_1;          /*                            */
    char    reserve_2[11];     /*                            */
    char    reserve_3[11];     /*                            */
}

/*****
/* Record type : datar_str
/* Description : structure of the datar
*****/

record datar_str
{
    long    cf;                /* certainty factor          */
    double  date_create;       /*                            */

```

```

        long   read_action;          /* action on read          */
        double double_1;             /*                          */
        char   reserve_2[11];        /*                          */
        char   reserve_3[11];        /*                          */
    }

/*****
/* Set type      : noun_set                      */
/* Description : Search path for noun            */
*****/
    set noun_set
    {
        order descending;
        owner brain;
        member noun by cf;
    }

/*****
/* Set type      : datar set                      */
/* Description : Search path for datar record    */
*****/
    set datar_set
    {
        order descending;
        owner noun;
        member noun_datar by cf;
    }

/*****
/* Set type      : datar_noun set                */
/* Description : Search path for noun from datar */
*****/
    set datar_noun_set
    {
        order descending;
        owner datar;
        member noun_datar by cf;
    }

/*****
/* Set type      : noun_synonim_exp_set          */
/* Description : Search path for noun synonym explosion */
*****/
    set noun_synonim_exp_set
    {
        order descending;
        owner noun;
        member noun_synonim by cf;
    }

/*****
/* Set type      : noun_synonim_imp_set          */
/* Description : Search path for noun synonym implosion */
*****/
    set noun_synonim_imp_set
    {
        order descending;
        owner noun;
        member noun_synonim by cf;
    }

/*****
/* Set type      : noun_exp_set                  */
*****/

```

```

/* Description : Search path for noun explosion */
/*****
    set noun_exp_set
    {
        order descending;
        owner noun;
        member noun_str by cf;
    }
*****/
/* Set type      : noun_imp_set */
/* Description : Search path for noun record from noun_str */
/*****
    set noun_imp_set
    {
        order descending;
        owner noun;
        member noun_str by cf;
    }
*****/
/* Set type      : datar_exp_set */
/* Description : Search path for datar explosion */
/*****
    set datar_exp_set
    {
        order descending;
        owner datar;
        member datar_str by cf;
    }
*****/
/* Set type      : datar_imp_set */
/* Description : Search path for datar record from datar_str */
/*****
    set datar_imp_set
    {
        order descending;
        owner datar;
        member datar_str by cf;
    }
*****/
/* Set type      : action_before_exp set */
/* Description : Search path for action_before from noun */
/*****
    set action_before_exp_set
    {
        order descending;
        owner noun;
        member action_before by cf;
    }
*****/
/* Set type      : action_before_imp set */
/* Description : Search path for action_before from noun */
/*****
    set action_before_imp_set
    {
        order descending;
        owner noun;
        member action_before by cf;
    }
*****/

```

```

    }
/*****
/* Set type      : action_after_exp set
/* Description : Search path for action_after from noun
*****/
    set action_after_exp_set
    {
        order descending;
        owner noun;
        member action_after by cf;
    }
/*****
/* Set type      : action_after_imp set
/* Description : Search path for action_after from noun
*****/
    set action_after_imp_set
    {
        order descending;
        owner noun;
        member action_after by cf;
    }
/*****
/* Set type      : datar_tabl_set
/* Description : Search path for datar_tabl from datar
*****/
    set datar_tabl_set
    {
        order descending;
        owner datar;
        member datar_tabl by cf;
    }
/* 1000-0-00-00-00 */
}
/*****
/* End of Schema: Brain Agenda
*****/

```

Figure 17. The RAIMA® Database Data Definition Language Schema for BrainAgenda®.

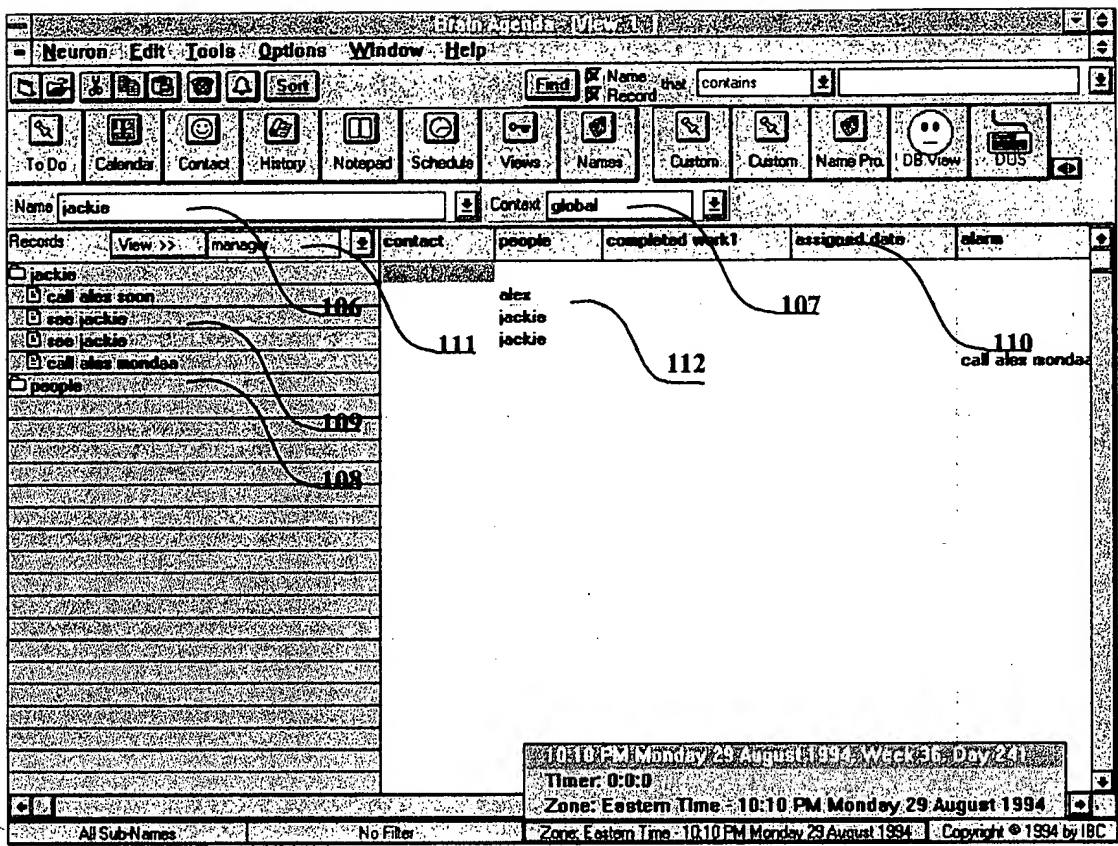


Figure 18. Two dimensional query results of the basic retrieval algorithm and elements of the spreadsheet interface. The Name and Context Combo is displayed.